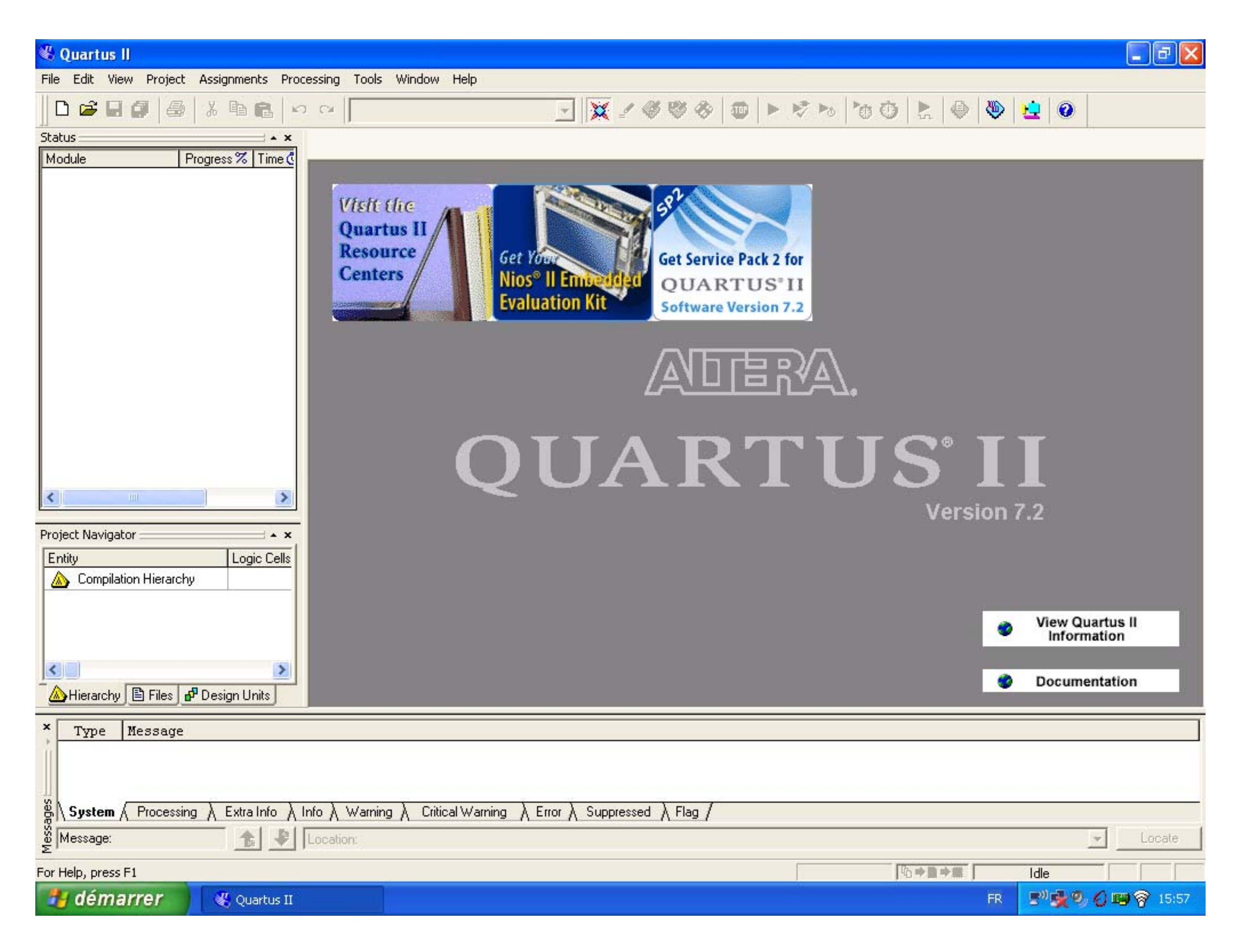
Manuel d'utilisation de Quartus II

1- Présentation

Ce document a pour but de vous initier à l'utilisation du logiciel Quartus II de la société Altéra ; les informations que vous trouverez dans ce document vous permettront de démarrer dans la création d'un projet. Elles ne constituent en rien une documentation complète et nous vous conseillons de consulter l'aide en ligne ou de parcourir le site Altéra pour une plus ample connaissance des outils logiciels.

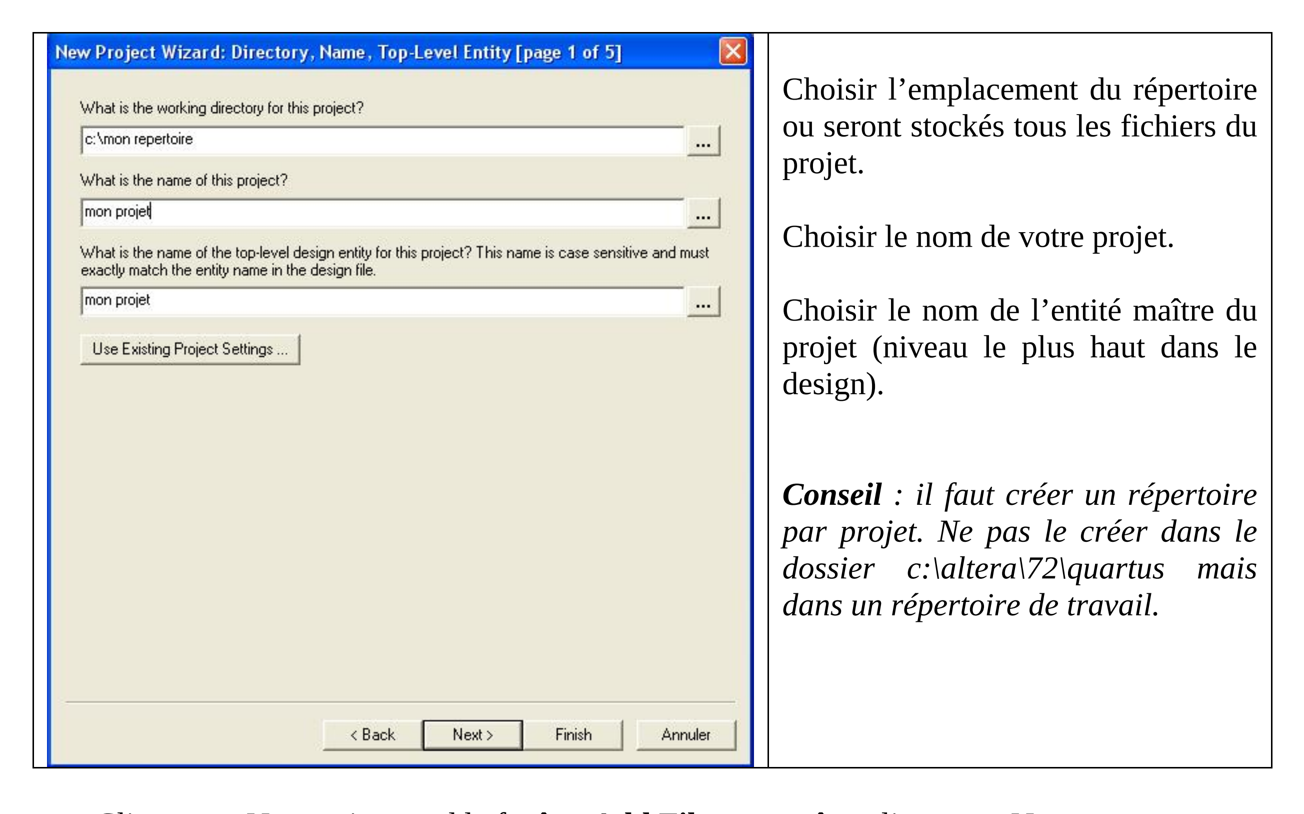


Quartus est un logiciel proposé par la société Altéra, permettant la gestion complète d'un flot de conception CPLD ou FPGA. Ce logiciel permet de faire une saisie graphique ou une saisie texte (description VHDL) d'en réaliser une simulation, une synthèse et une implémentation sur cible reprogrammable.

***Conseil** : Bien que ce ne soit pas obligatoire, nous vous recommandons d'utiliser une description schématique au niveau le plus élevé de votre hiérarchie, ceci afin d'avoir une vue d'ensemble de votre projet. Ensuite, vous pourrez adjoindre des modules décrits soit en schématique, soit en VHDL.*

2- Création d'un nouveau projet

File → New Project Wizard → OK



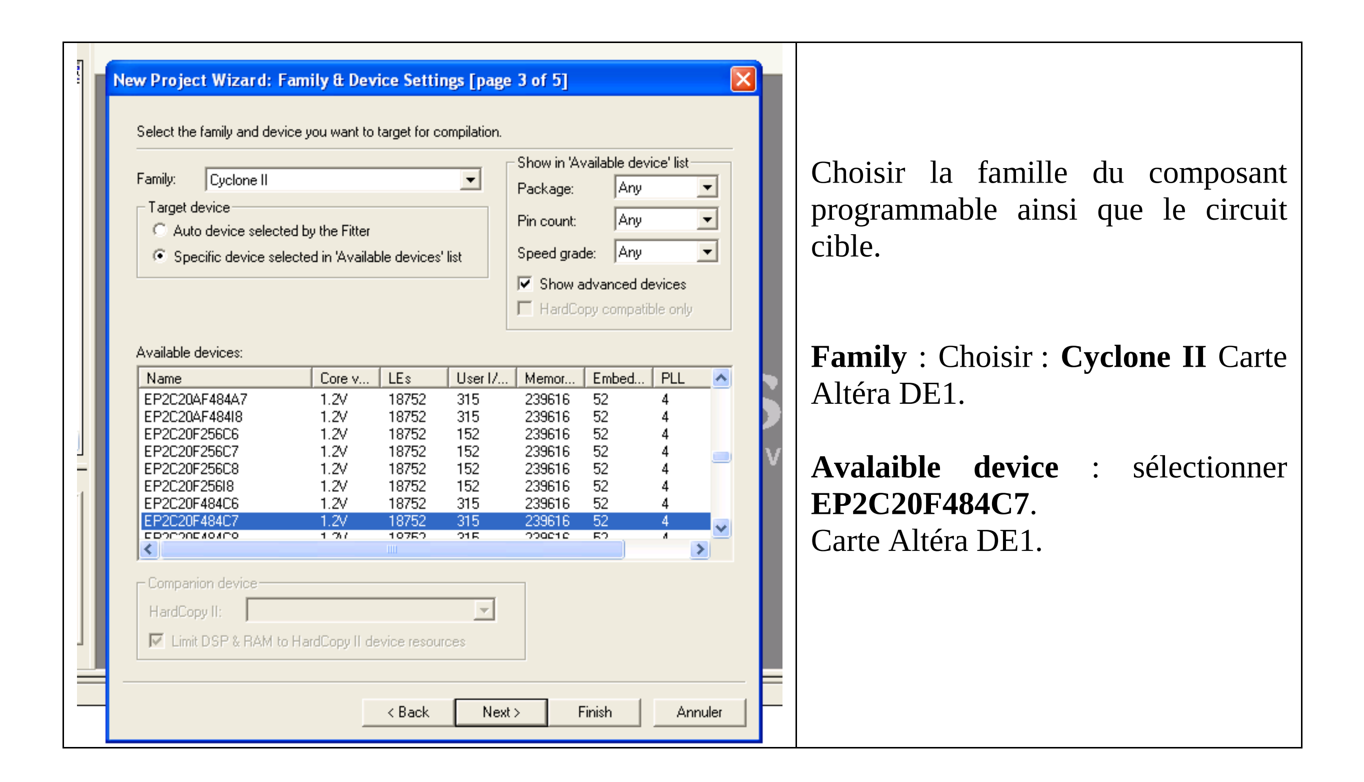
Choisir l'emplacement du répertoire ou seront stockés tous les fichiers du projet.

Choisir le nom de votre projet.

Choisir le nom de l'entité maître du projet (niveau le plus haut dans le design).

***Conseil** : il faut créer un répertoire par projet. Ne pas le créer dans le dossier c:\altera\72\quartus mais dans un répertoire de travail.*

Cliquer sur **Next** puis quand la fenêtre **Add Files** apparaît recliquer sur **Next**.

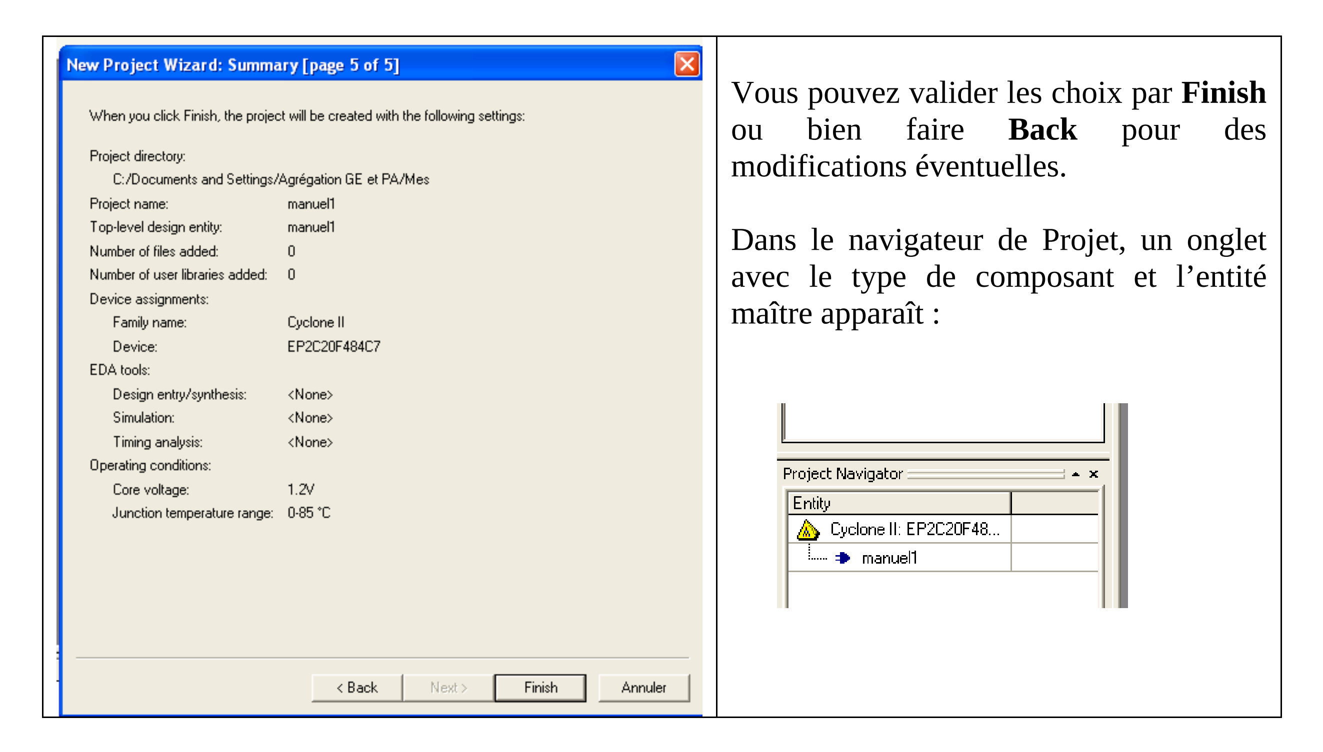


Choisir la famille du composant programmable ainsi que le circuit cible.

Family : Choisir : **Cyclone II** Carte Altéra DE1.

Avalaible device : sélectionner **EP2C20F484C7**.
Carte Altéra DE1.

Next → Quand la fenêtre **EDA Tool Settings** apparaît cliquer sur **Next →** une fenêtre récapitulative apparaît :



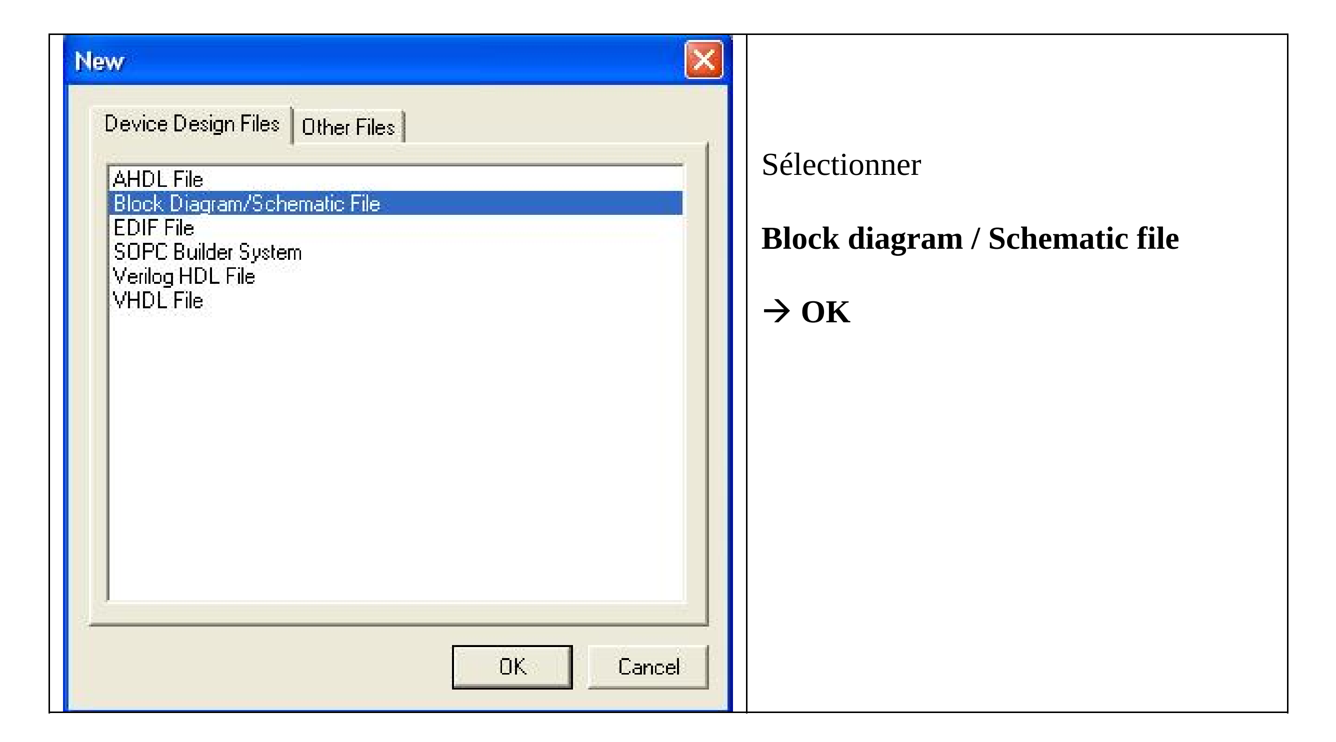
Vous pouvez valider les choix par **Finish** ou bien faire **Back** pour des modifications éventuelles.

Dans le navigateur de Projet, un onglet avec le type de composant et l'entité maître apparaît :

3- Saisie d'un projet

3.1- Création d'un schéma

File → New



Sélectionner

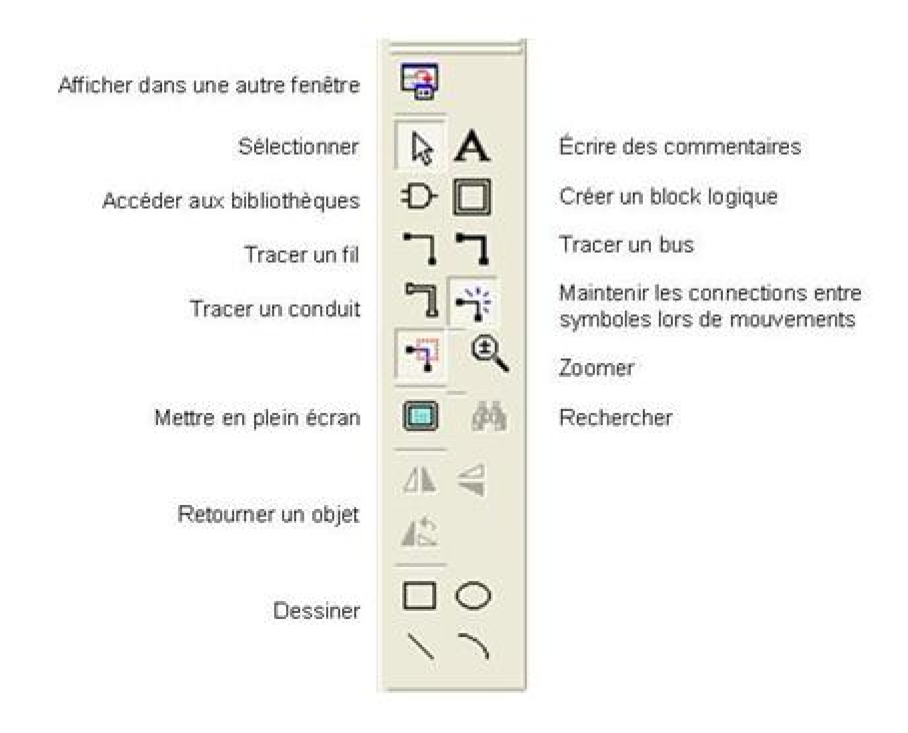
Block diagram / Schematic file

→ OK

File → Save as.

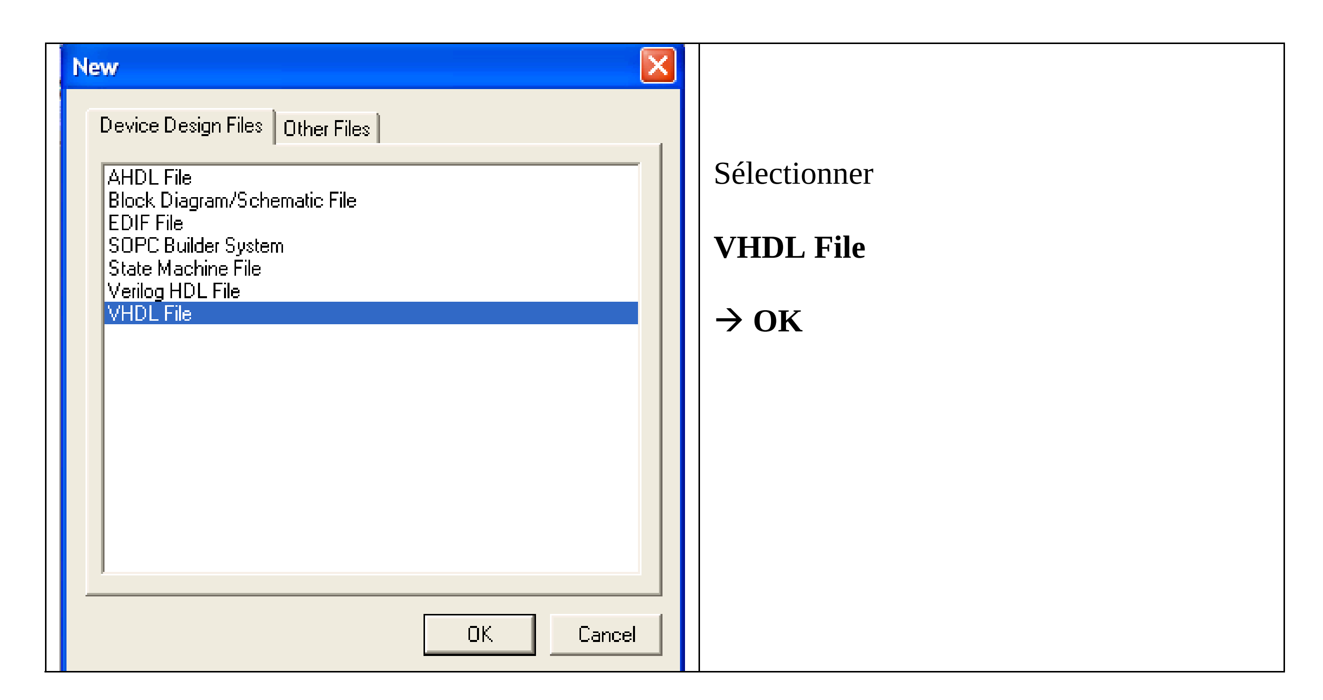
***Conseil** : Une feuille blanche se crée intitulée Block1.bdf. On prendra soin de sauver cette feuille sous le nom de l'entité maître, car c'est maintenant cette feuille de saisie graphique qui a la hiérarchie la plus haute dans le projet.*

Utilisation de la boite à outils :



3.2- Création d'un fichier VHDL

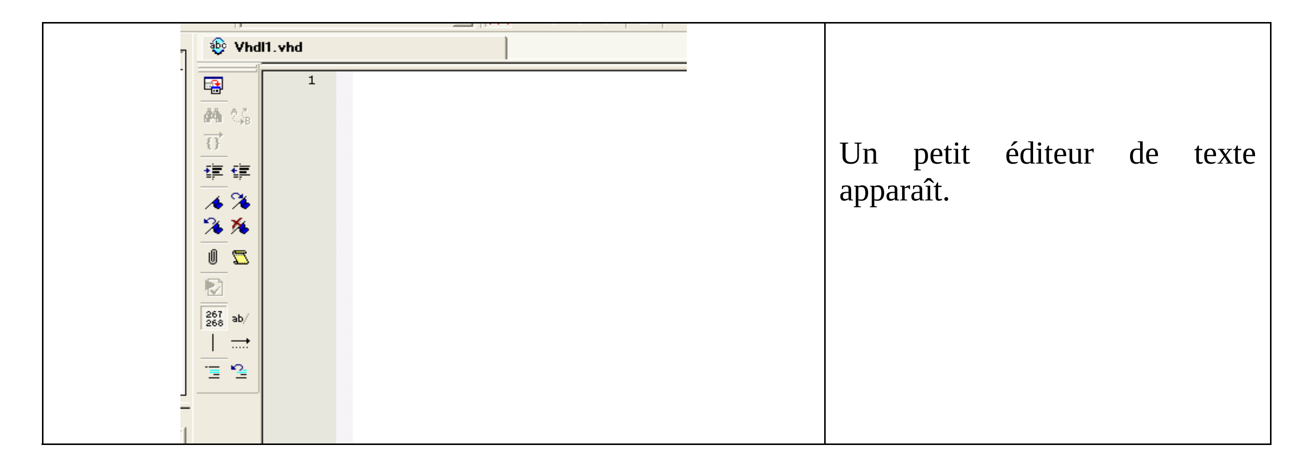
File → New



Sélectionner

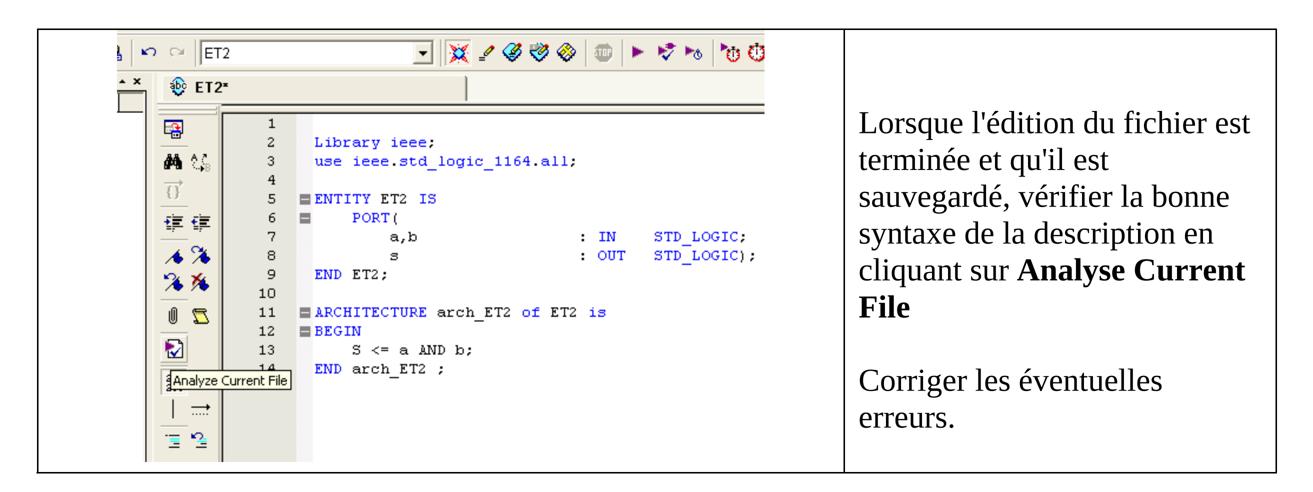
VHDL File

→ OK



Un petit éditeur de texte apparaît.

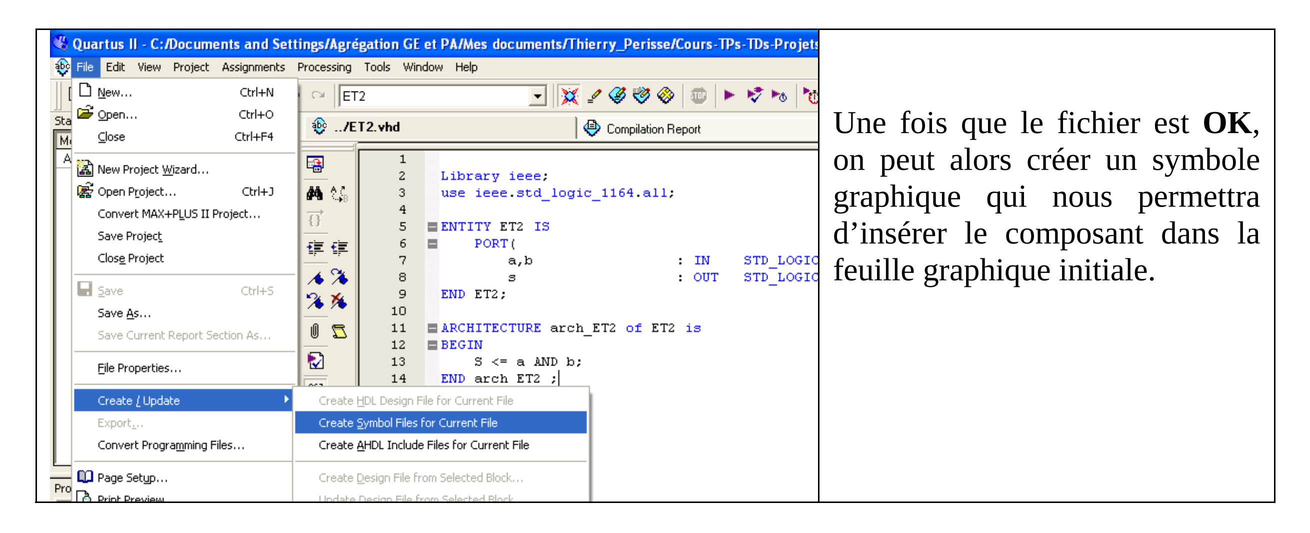
Une fois le code VHDL saisie, il convient de le sauver (**File** puis **Save As**) puis d'en vérifier la syntaxe.
Conseil *: Il est important de sauver le fichier sous le même nom que l'entité. Bien que cela ne soit pas indispensable comme sous MaxplusII, cela évite des intersections d'entité entre fichiers.*



Lorsque l'édition du fichier est terminée et qu'il est sauvegardé, vérifier la bonne syntaxe de la description en cliquant sur **Analyse Current File**

Corriger les éventuelles erreurs.

3.3- Création d'un symbole



Une fois que le fichier est **OK**, on peut alors créer un symbole graphique qui nous permettra d'insérer le composant dans la feuille graphique initiale.

4- Compilation

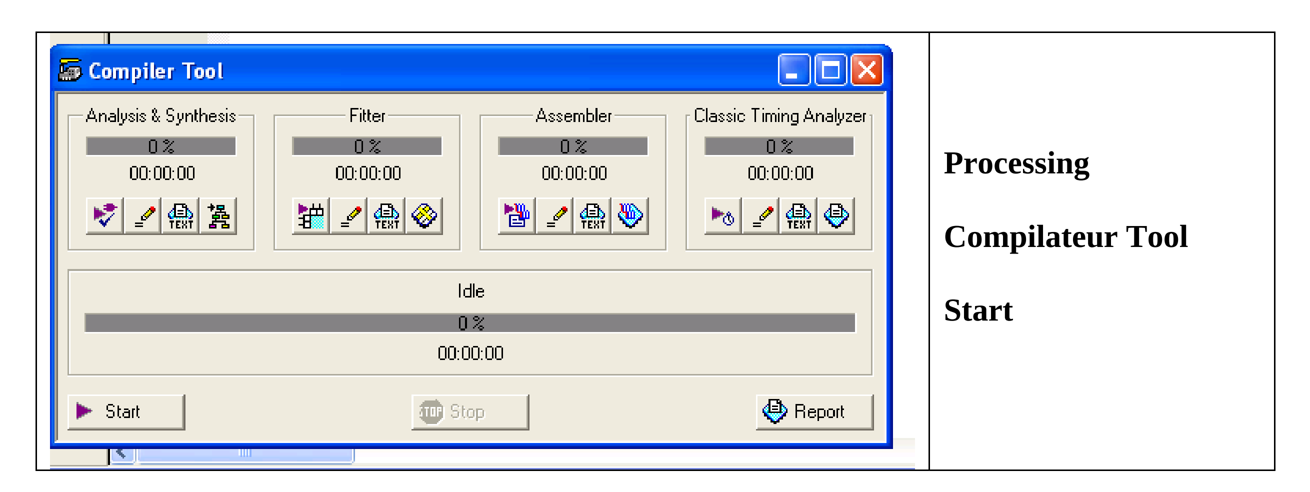
Durant la compilation, Quartus va réaliser 4 étapes :

a- La transformation des descriptions graphiques et textuelles en un schéma électronique à base de portes et de registres : c'est la ***synthèse logique***.

b- L'étape de Fitting (ajustement) consiste à voir comment les différentes portes et registres (produit par la synthèse logique) peuvent être placés en fonction des ressources matérielles du circuit cible (EP2C20F484C7) : c'est la ***synthèse physique***.

c- L'assemblage consiste à produire les fichiers permettant la programmation du circuit. Ce sont des fichiers au format Programmer Object Files (.pof), SRAM Object Files (.sof), Hexadécimal (Intel-Format) Ouput Files (.hexout), Tabular Text Files (.ttf), et Raw Binary Files (.rbf).
Dans notre cas, nous utiliserons toujours le format SOF pour les FPGA et le format POF pour les CPLD.

d- L'analyse temporelle permet d'évaluer les temps de propagation entre les portes et le long des chemins choisis lors du fitting.

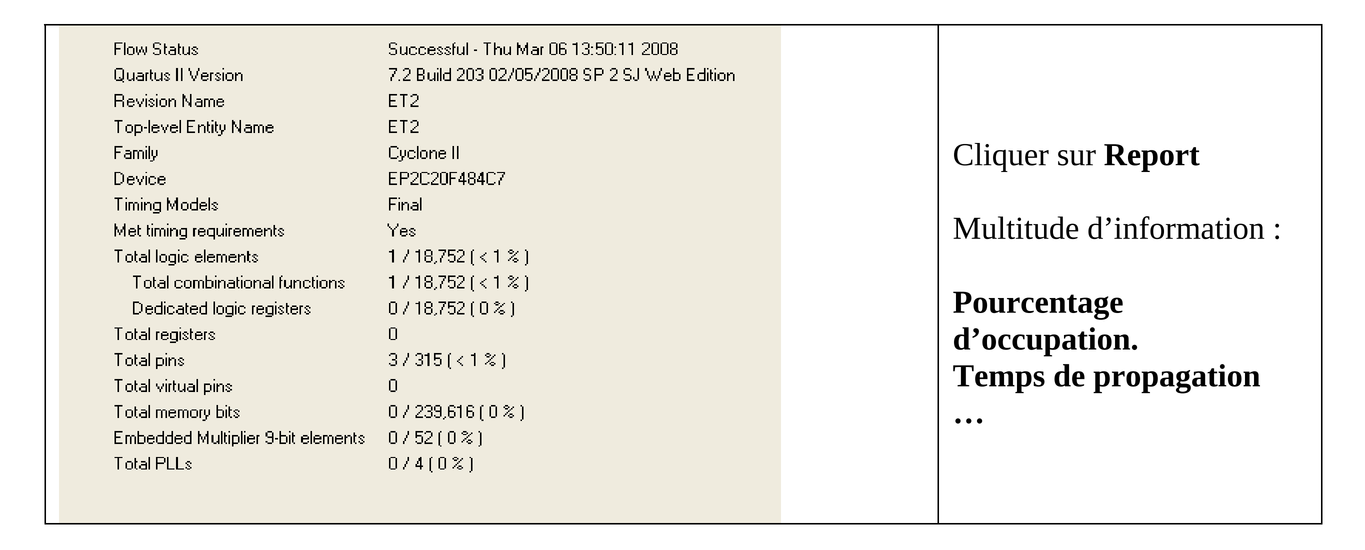


Processing

Compilateur Tool

Start

Normalement, il ne doit pas y avoir d'erreur. Si ce n'est pas le cas, vérifier dans la zone **Processing** (en bas où s'affichent les messages) la source du problème.

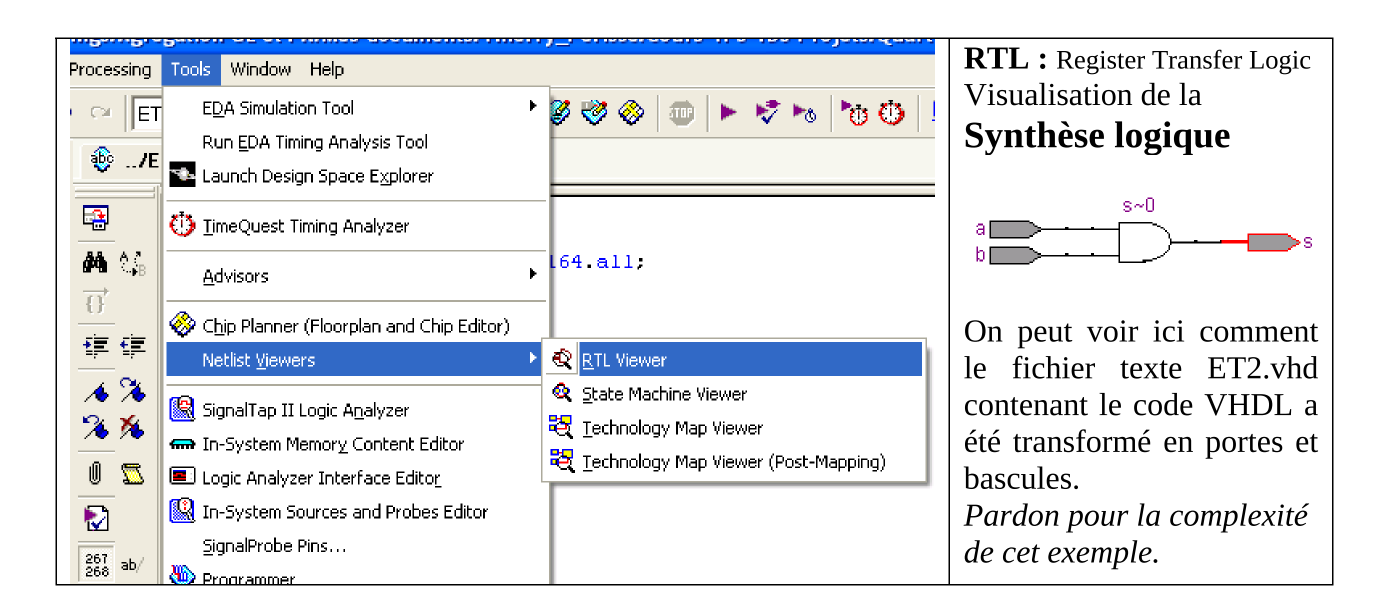


Flow Status	Successful - Thu Mar 06 13:50:11 2008
Quartus II Version	7.2 Build 203 02/05/2008 SP 2 SJ Web Edition
Revision Name	ET2
Top-level Entity Name	ET2
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	1 / 18,752 (< 1 %)
Total combinational functions	1 / 18,752 (< 1 %)
Dedicated logic registers	0 / 18,752 (0 %)
Total registers	0
Total pins	3 / 315 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

Cliquer sur **Report**

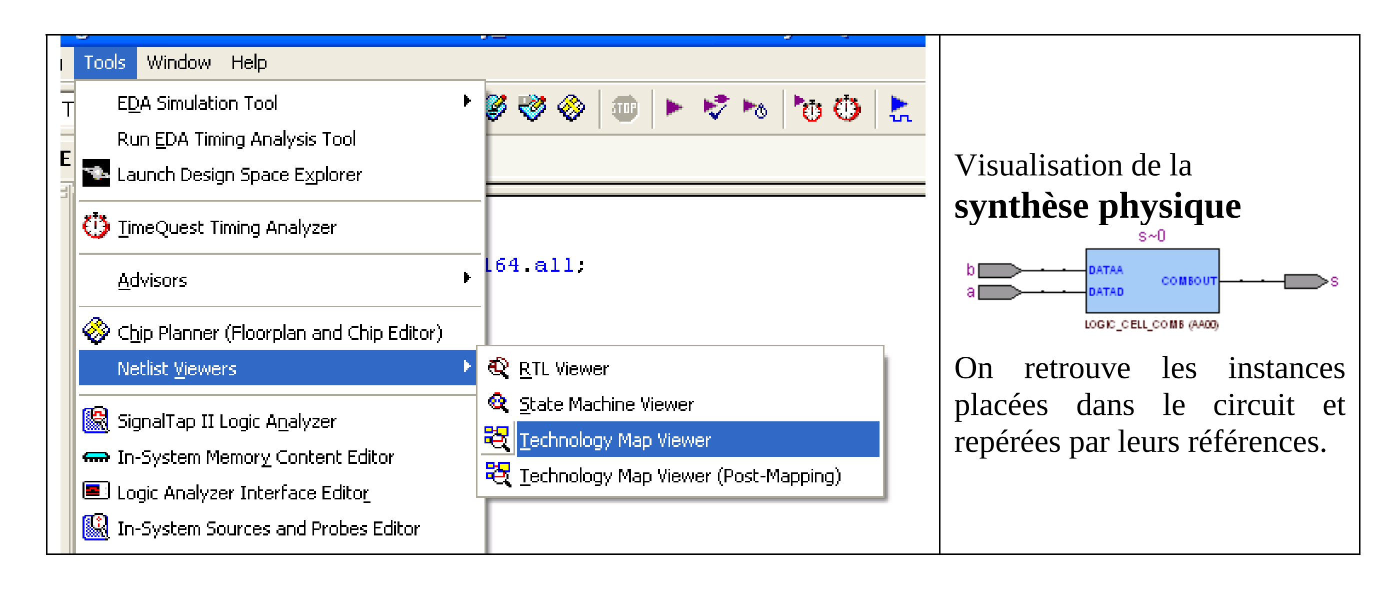
Multitude d'information :

Pourcentage d'occupation.
Temps de propagation
…



RTL : Register Transfer Logic
Visualisation de la
Synthèse logique

On peut voir ici comment le fichier texte ET2.vhd contenant le code VHDL a été transformé en portes et bascules.
Pardon pour la complexité de cet exemple.



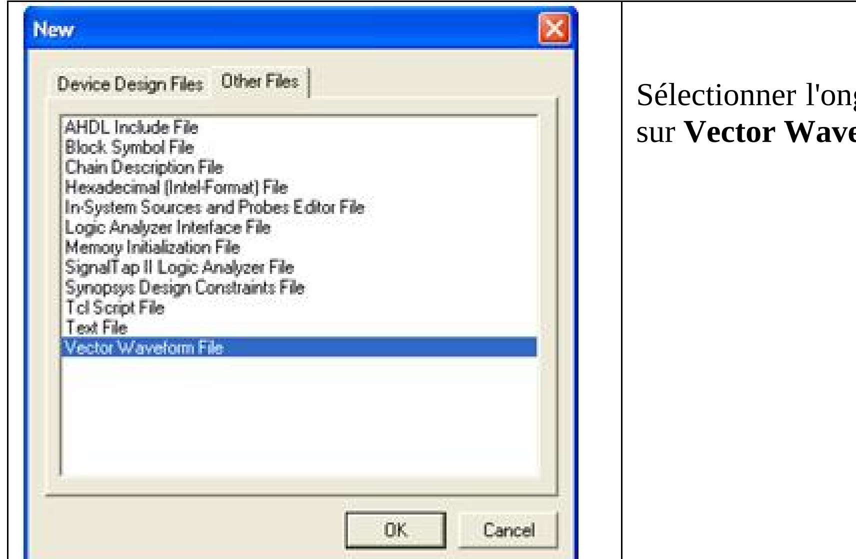
Visualisation de la
synthèse physique

On retrouve les instances placées dans le circuit et repérées par leurs références.

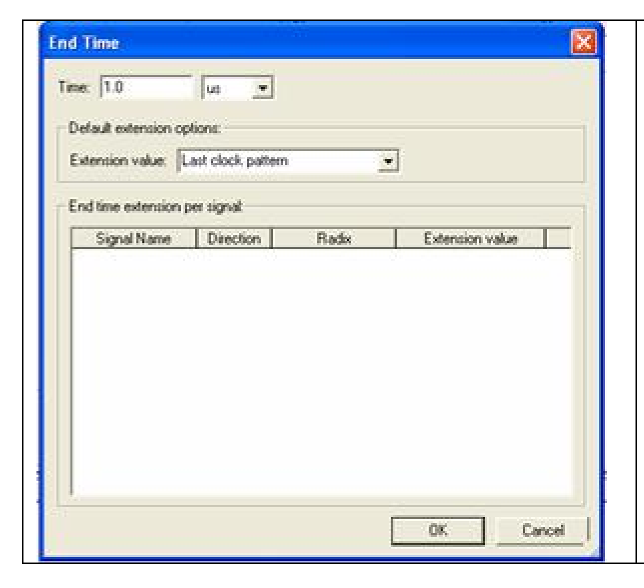
5- Simulation d'un circuit

- La partie du circuit à simuler doit être munie de pins d'entrée/sortie.
- Elle doit aussi se trouver au niveau le plus élevé de la hiérarchie. Si ce n'est pas le cas, pour l'y mettre : dans le "**Project Navigator**", cliquer avec le bouton droit de la souris sur le nom du fichier, puis sur **Set as Top-Level Entity**.
- Il faut également vérifier qu'il n'y ait pas d'erreur dans le circuit en cliquant sur **Processing** puis sur **Start**, et enfin sur **Start Analysis & Elaboration**.
- Le circuit étant prêt, il faut maintenant créer le fichier contenant les informations sur les signaux à appliquer sur les entrées du composant et la liste des signaux que l'on veut analyser.

Cliquer sur **File** puis sur **New**

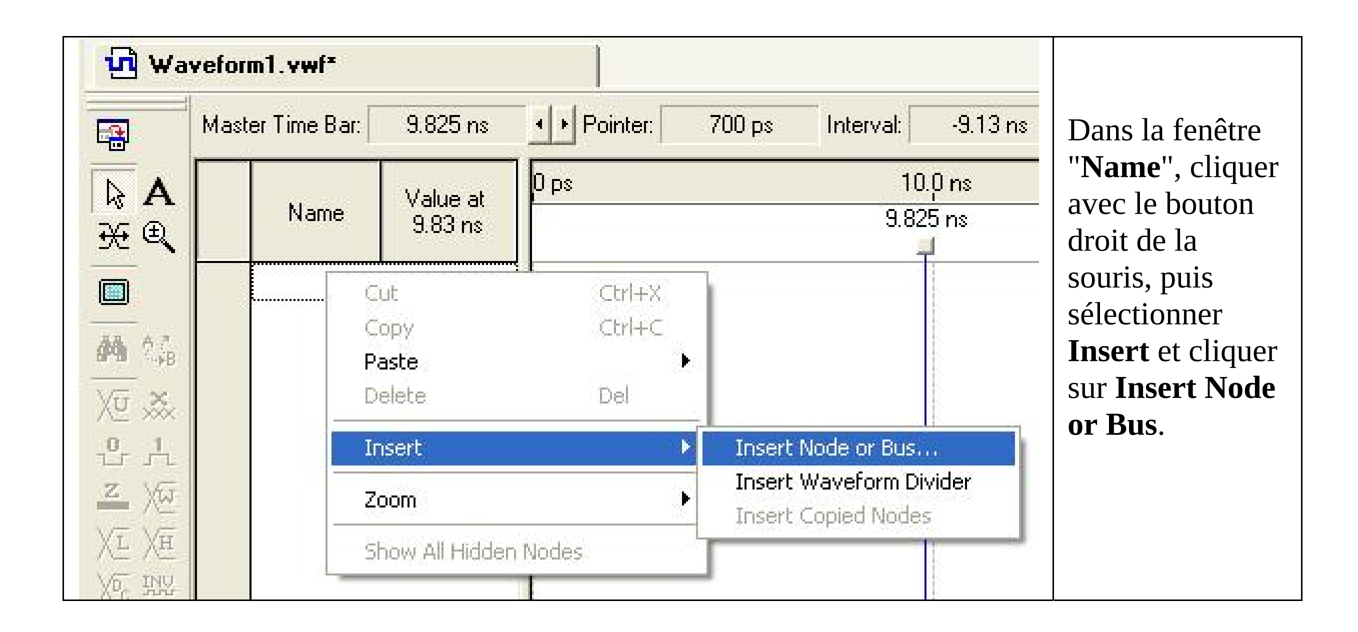


Sélectionner l'onglet "**Other Files**" et cliquer sur **Vector Waveform File**.

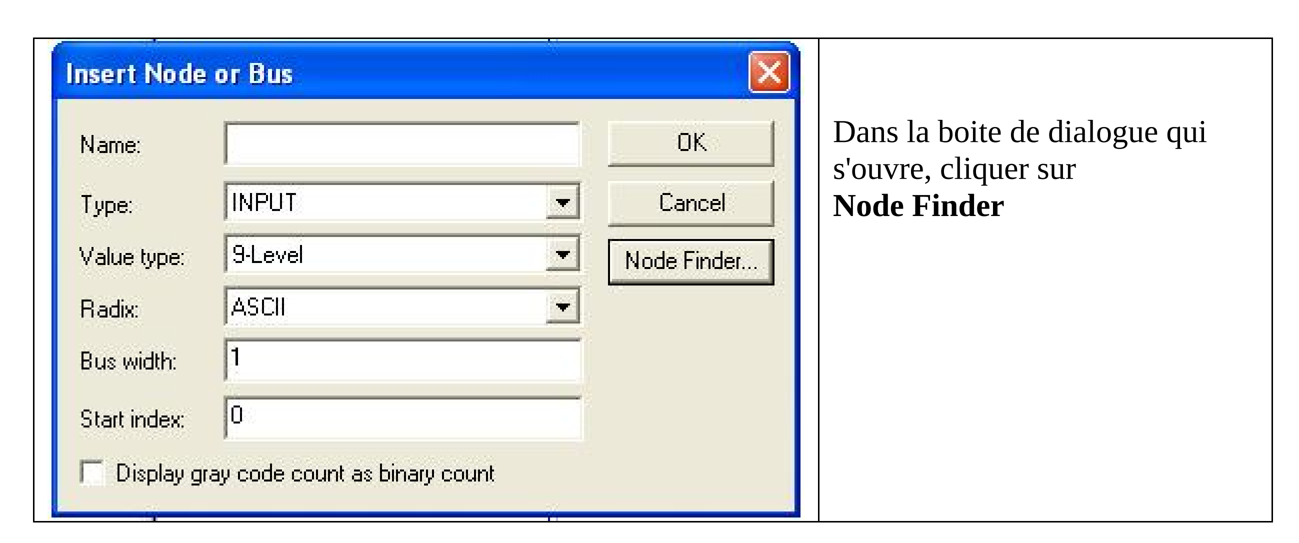


Par défaut, la durée de simulation est de **1 µs**. Pour la modifier, cliquer sur **Edit**, puis **End Time**.
Une fois la durée modifiée, cliquer sur **OK**.

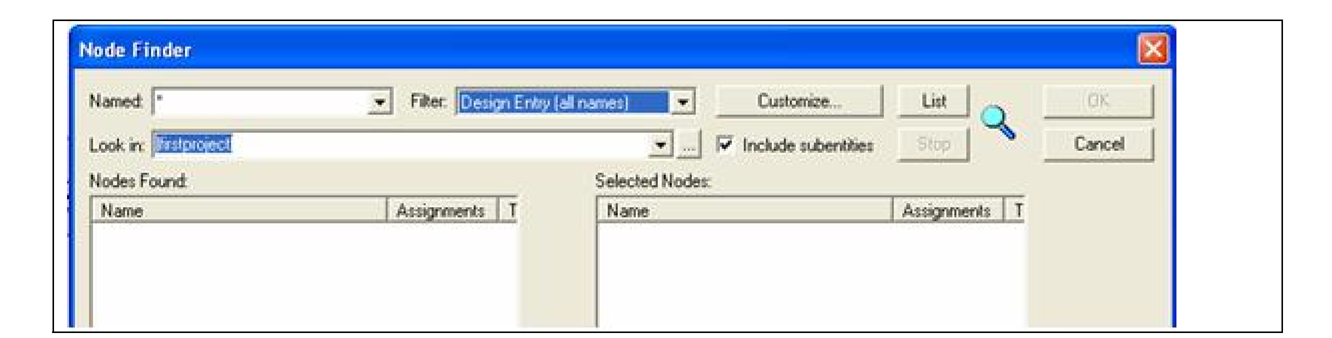
Sauvegarder le fichier sous son nom définitif avec son extension **(.vwf)** en cliquant sur **File** puis **Save As**.



Dans la fenêtre "**Name**", cliquer avec le bouton droit de la souris, puis sélectionner **Insert** et cliquer sur **Insert Node or Bus**.



Dans la boite de dialogue qui s'ouvre, cliquer sur **Node Finder**



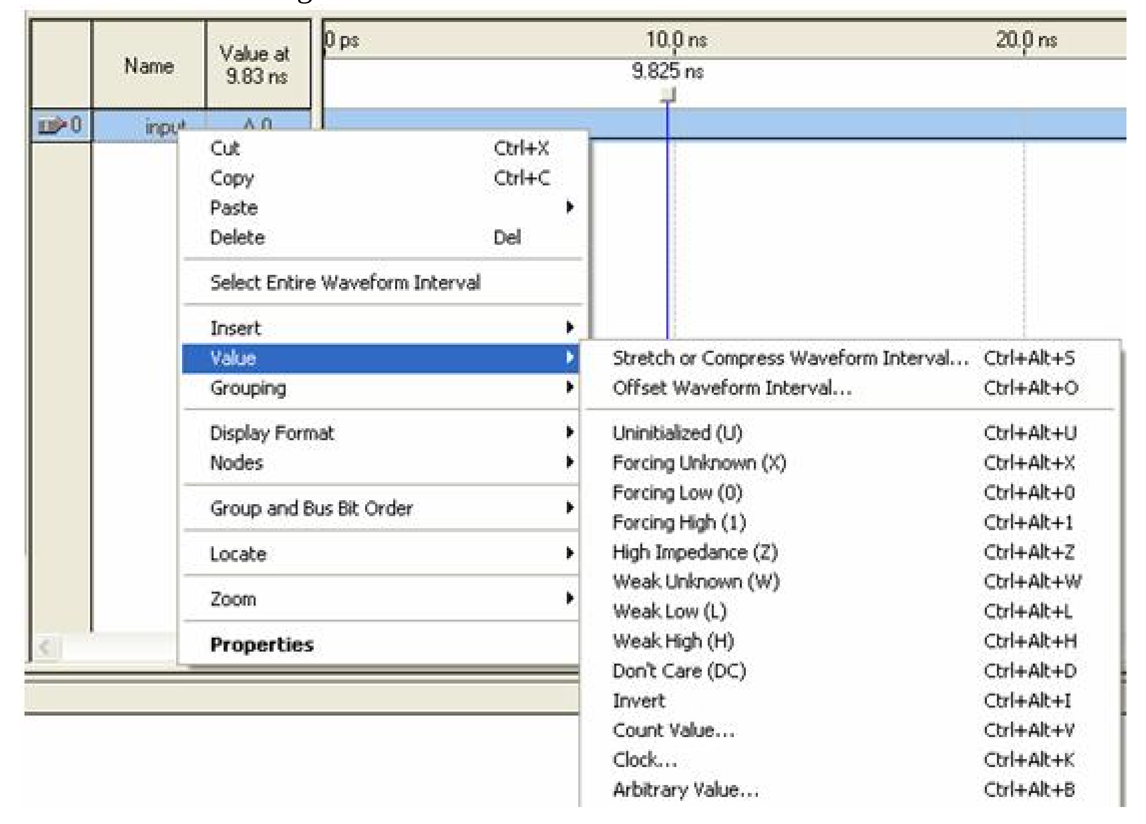
Dans la catégorie "**Filter**", choisir **all names**. Cliquer ensuite sur le bouton **List**.

Ajouter les signaux souhaités dans la fenêtre **Selected Nodes**, en cliquant sur > .

Cliquer sur **OK** pour fermer les différentes fenêtres et revenir à l'éditeur de signaux.

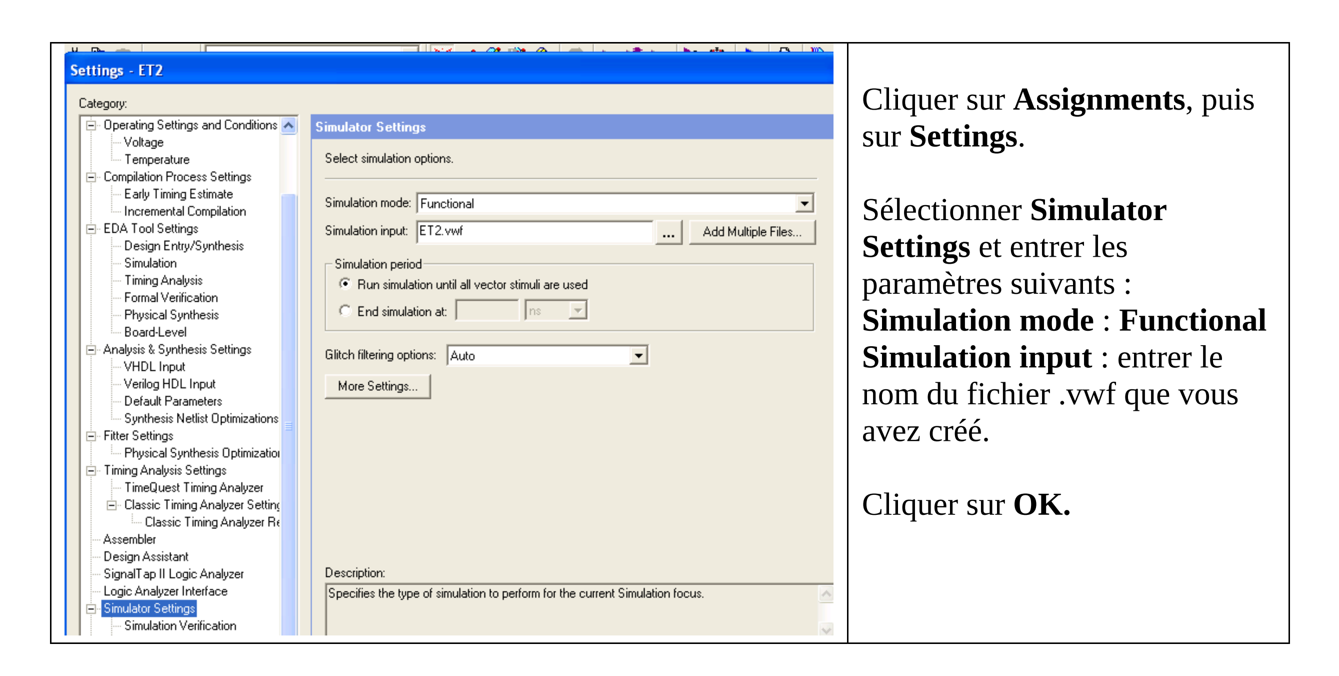
Afin de simuler le design, il convient de lui injecter des stimuli. Lorsque ces stimuli sont générés à partir d'un fichier on dit que l'on utilise un **fichier de Bench**.

Cliquer avec le bouton droit de la souris sur le nom d'un signal, sélectionner **Value**, puis choisir la valeur du signal dans le menu.



Il est possible d'effectuer la même opération sur une partie seulement d'un signal en sélectionnant une zone dans la partie "chronogramme". Il faut pour cela maintenir le bouton gauche de la souris appuyé en déplaçant le curseur.
Lorsque tous les signaux d'entrées sont définis, sauvegarder le fichier.

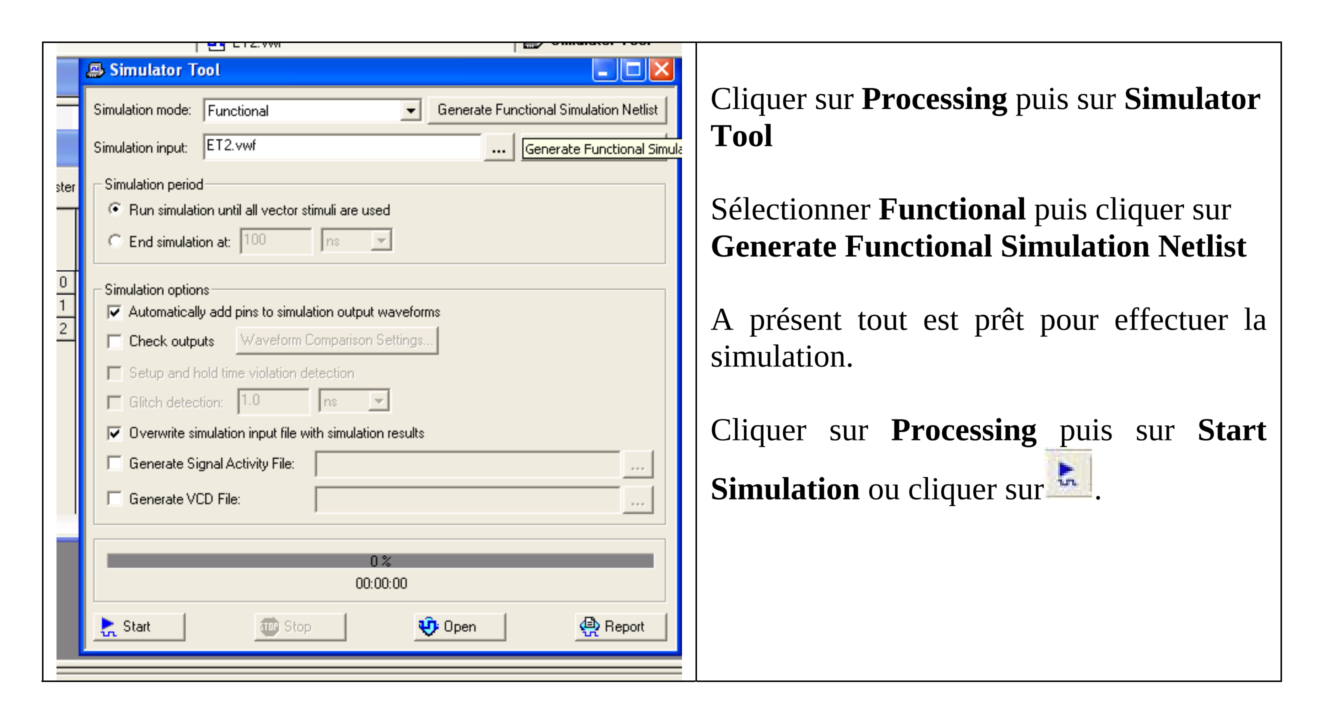
5.1- Simulation Fonctionnelle



Cliquer sur **Assignments**, puis sur **Settings**.

Sélectionner **Simulator Settings** et entrer les paramètres suivants :
Simulation mode : **Functional**
Simulation input : entrer le nom du fichier .vwf que vous avez créé.

Cliquer sur **OK.**

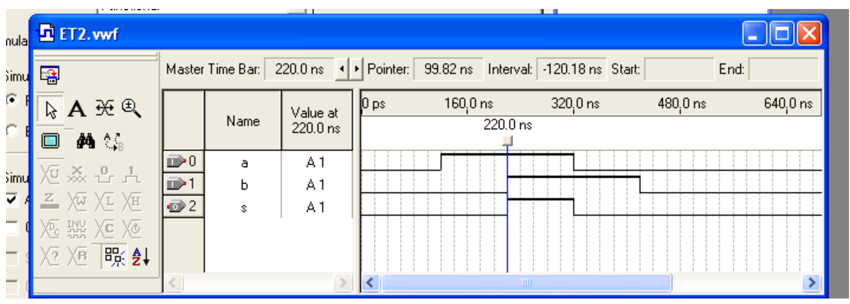


Cliquer sur **Processing** puis sur **Simulator Tool**

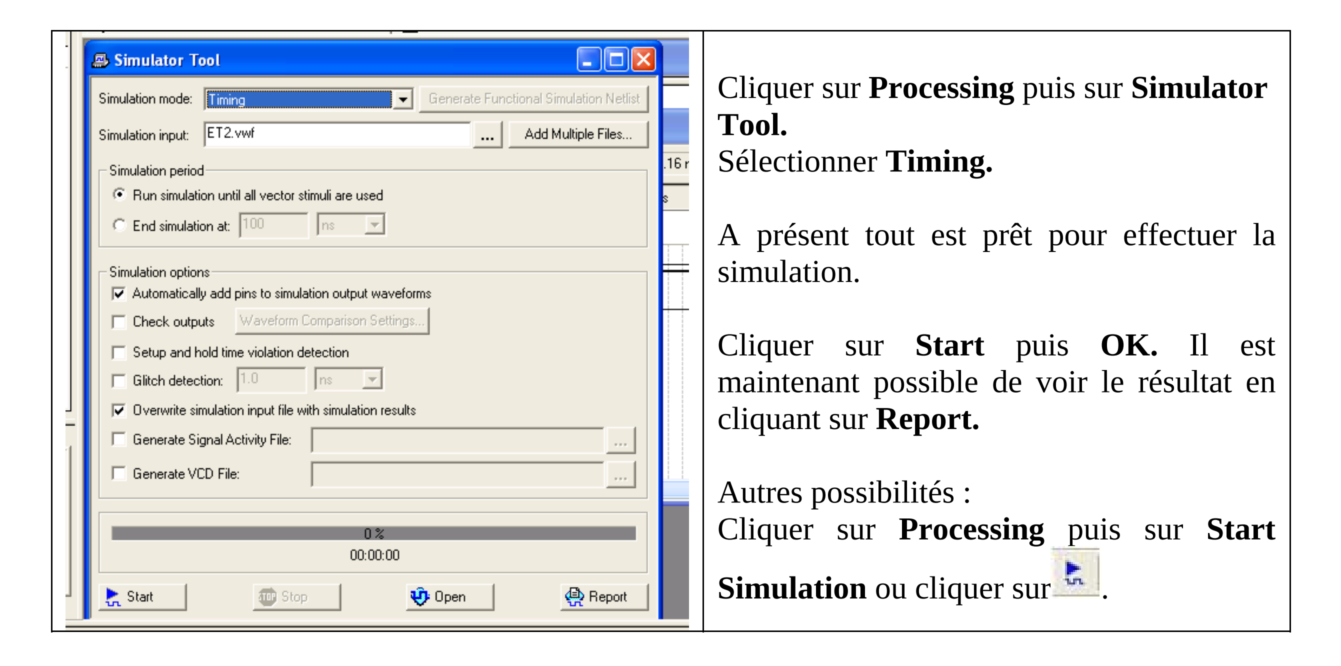
Sélectionner **Functional** puis cliquer sur **Generate Functional Simulation Netlist**

A présent tout est prêt pour effectuer la simulation.

Cliquer sur **Processing** puis sur **Start Simulation** ou cliquer sur .



5.2- Simulation Temporelle

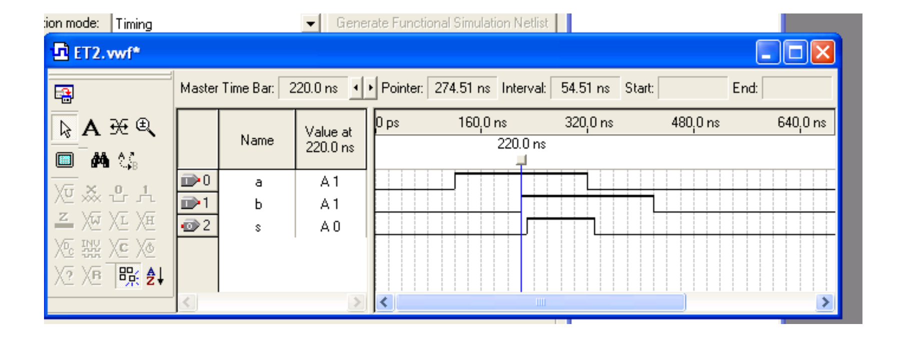


Cliquer sur **Processing** puis sur **Simulator Tool.**
Sélectionner **Timing.**

A présent tout est prêt pour effectuer la simulation.

Cliquer sur **Start** puis **OK.** Il est maintenant possible de voir le résultat en cliquant sur **Report.**

Autres possibilités :
Cliquer sur **Processing** puis sur **Start Simulation** ou cliquer sur .


ion mode: Timing
Generate Functional Simulation Netlist
ET2.vwf*
Master Time Bar: 220.0 ns
Pointer: 274.51 ns
Interval: 54.51 ns
Start:
End:
Name
Value at 220.0 ns
0 a A 1
1 b A 1
2 s A 0
0 ps
160,0 ns
320,0 ns
480,0 ns
640,0 ns
220.0 ns

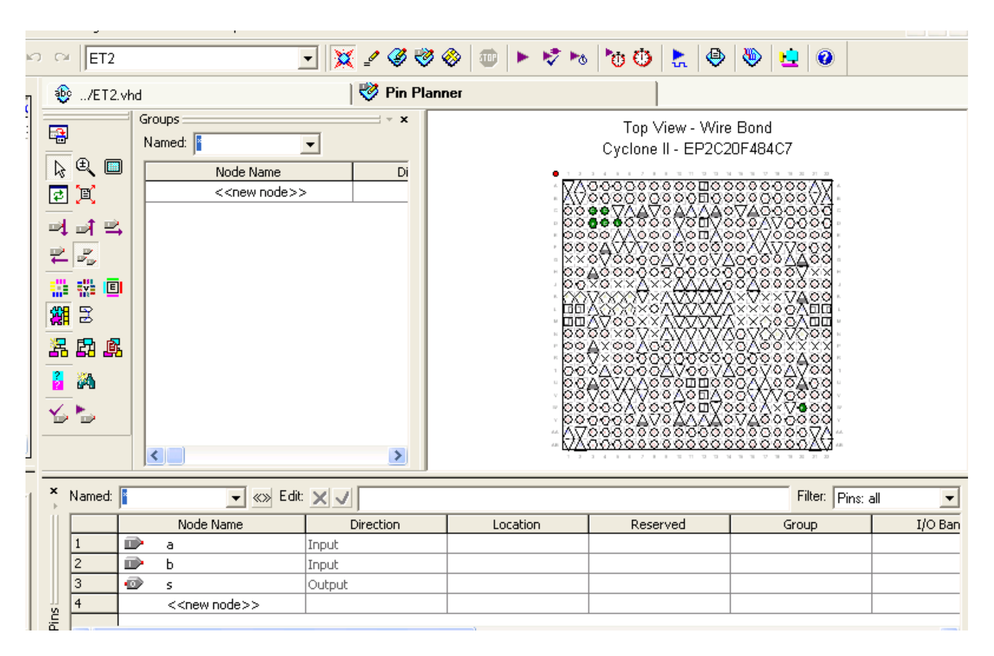
6- Programmation d'un circuit

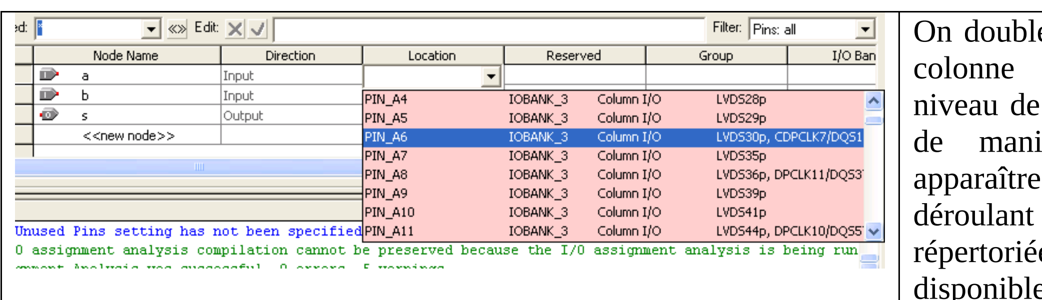
C'est l'étape ultime !

Pour cela, il faut assigner les pins d'entrées et de sorties du design aux broches du circuit physique.

6.1- Affectation des entrées et des sorties

Cliquer sur **Assignments** puis sur **pins**





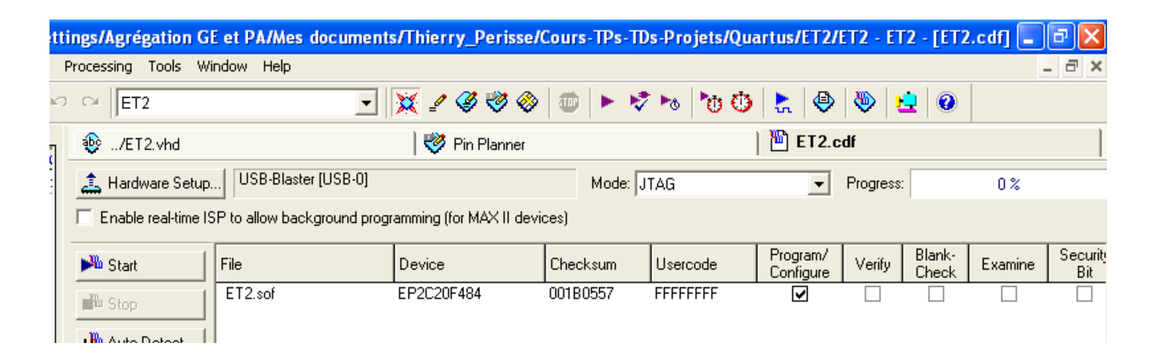
On double clique sur la colonne **location** au niveau de la pin voulue de manière à faire apparaître un menu déroulant où sont répertoriées les broches disponibles du circuit.

La liste des broches utilisables pour le FPGA et sortant sur les connecteurs est donnée dans le manuel de la carte DE1 d'Altéra.
Ne pas oublier de compiler avant la programmation.

6.2- Programmation du circuit

La programmation du circuit se fait via le protocole JTAG. Pour cela, vérifier que la connection entre le PC et la carte DE1 via le module USB-Blaster est opérationnelle.

Si tout est **ok** lancer le programmateur :
Cliquer sur **Tools** puis sur **Programmer.**



Vérifier que le fichier avec l'extension **.sof** est bien là (sélectionner le) et que la case **Program/Configure** est cochée, puis cliquer sur **Start**.

C'est fini !

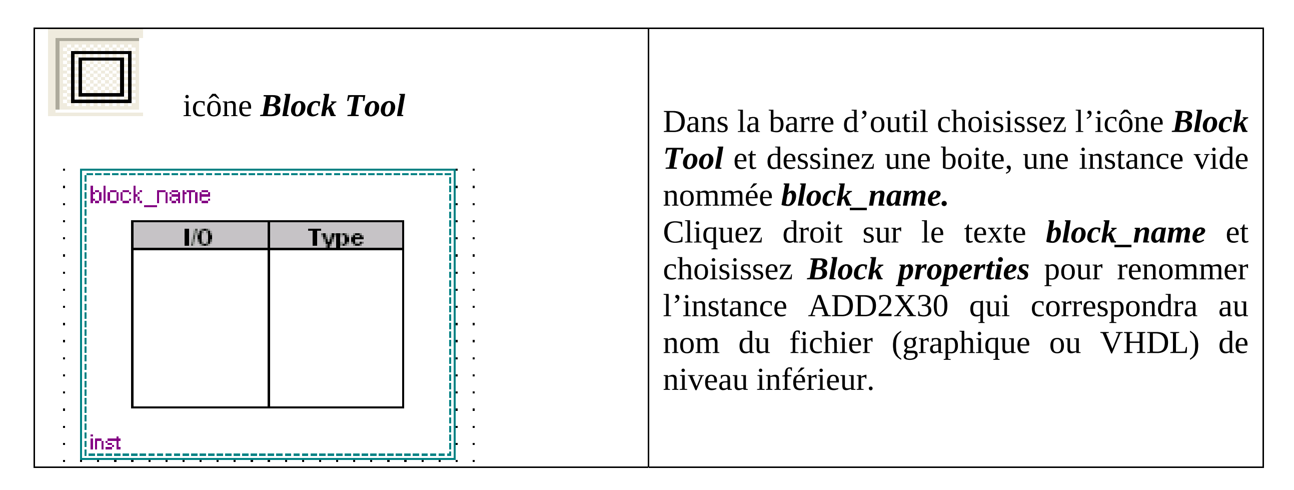
Il n'y a plus qu'à vérifier et à débugger.

7- Complément : Projet multi support, hiérarchisation

On désire réaliser un additionneur 3 bits en utilisant l'additionneur 1 bit déjà compilé.

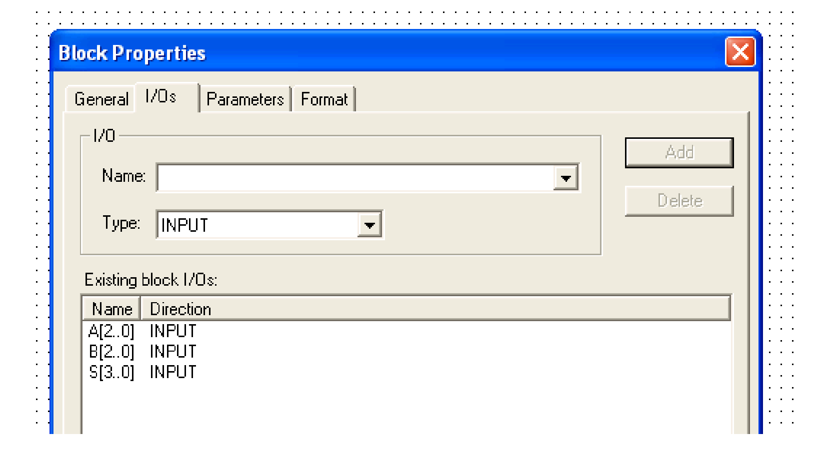
7.1 Création du projet

Définissez un nouveau projet nommé ADD2X3 et ouvrez une page graphique ADD2X3.bdf, cette page nous servira pour définir la structure hiérarchique haute de notre projet.

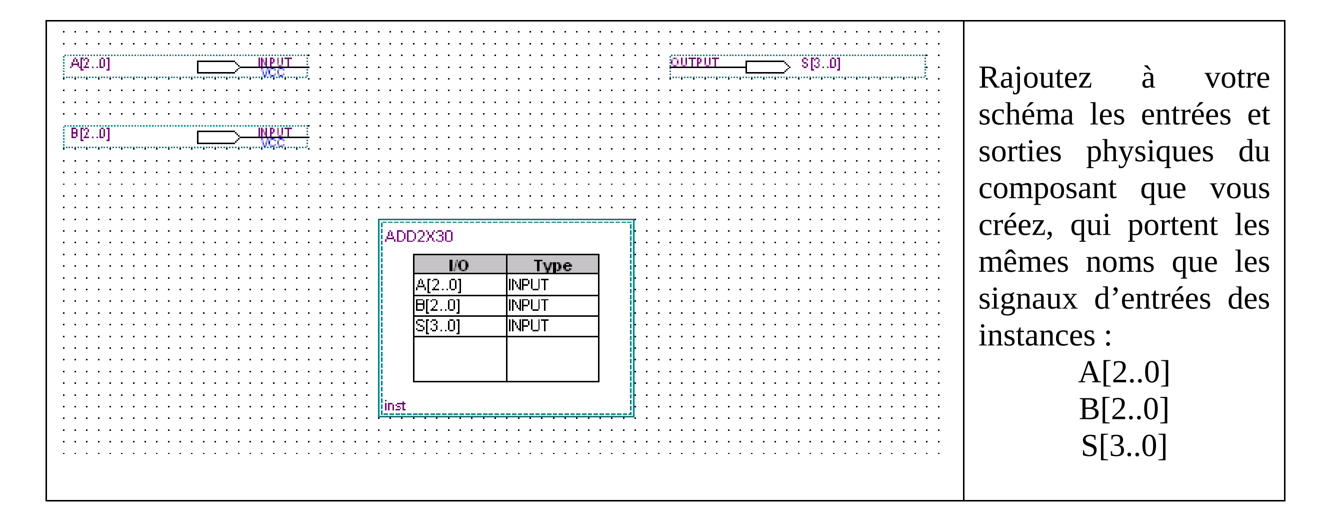


Dans la barre d'outil choisissez l'icône ***Block Tool*** et dessinez une boite, une instance vide nommée ***block_name.***
Cliquez droit sur le texte ***block_name*** et choisissez ***Block properties*** pour renommer l'instance ADD2X30 qui correspondra au nom du fichier (graphique ou VHDL) de niveau inférieur.

Nous allons maintenant définir les signaux d'entrées et de sortie de notre instance, pour cela cliquez sur le bouton droit et éditez les propriétés du block. Dans l'onglet I/Os, nous définirons les entrées et les sorties désirées en cliquant sur **Add** pour ajouter les différents signaux de notre première instance.

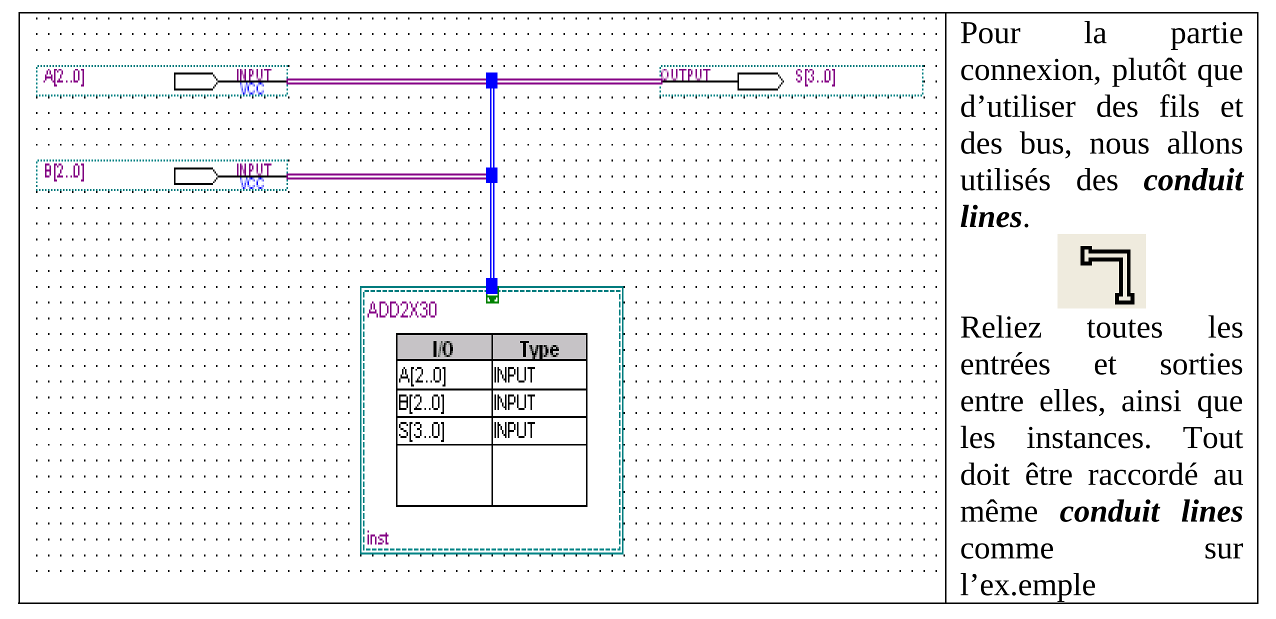


Une fois fini, cliquer sur **ok**, l'instance dans le schématic est automatiquement modifiée.



Rajoutez à votre schéma les entrées et sorties physiques du composant que vous créez, qui portent les mêmes noms que les signaux d'entrées des instances :

A[2..0]
B[2..0]
S[3..0]



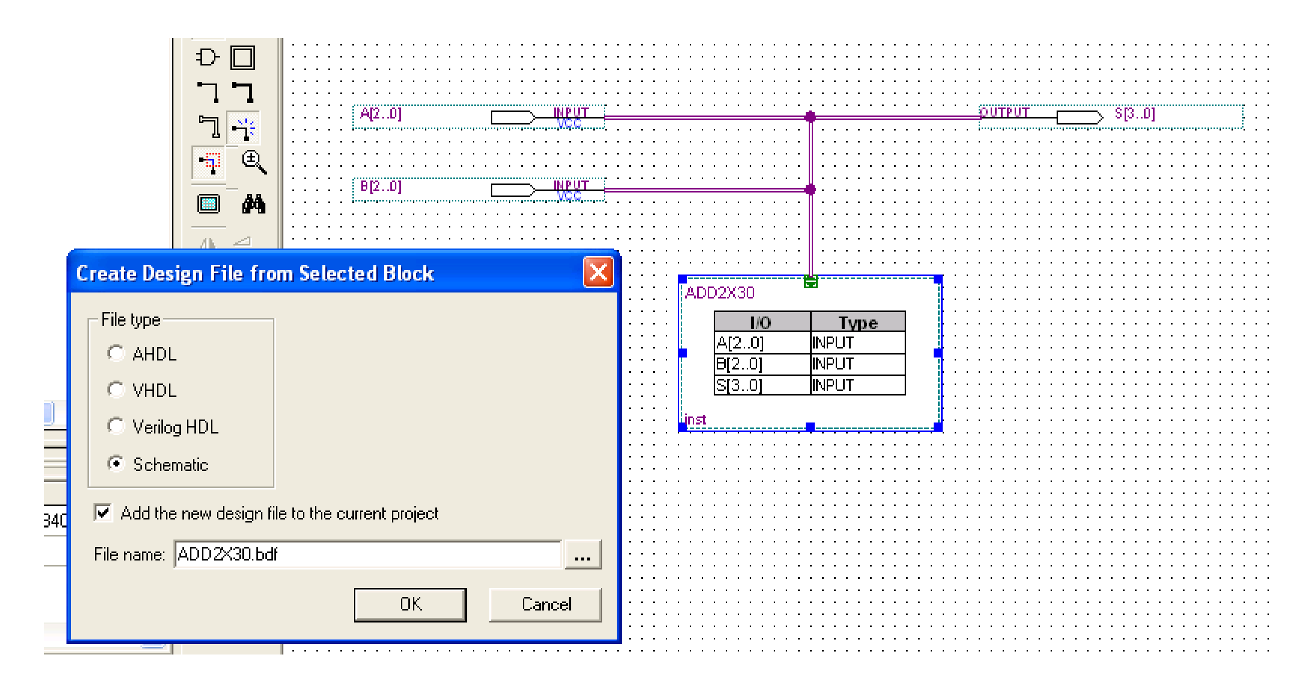
Pour la partie connexion, plutôt que d'utiliser des fils et des bus, nous allons utilisés des ***conduit lines***.

Reliez toutes les entrées et sorties entre elles, ainsi que les instances. Tout doit être raccordé au même ***conduit lines*** comme sur l'ex.emple

Avec le bouton droit, cliquez maintenant sur une partie du conduit line, et éditez les propriétés de celle-ci. Allez sur l'onglet signal, le logiciel a automatiquement associé les bons signaux aux bonnes instances. Attention ceci ne marche que si les noms des signaux coïncident.

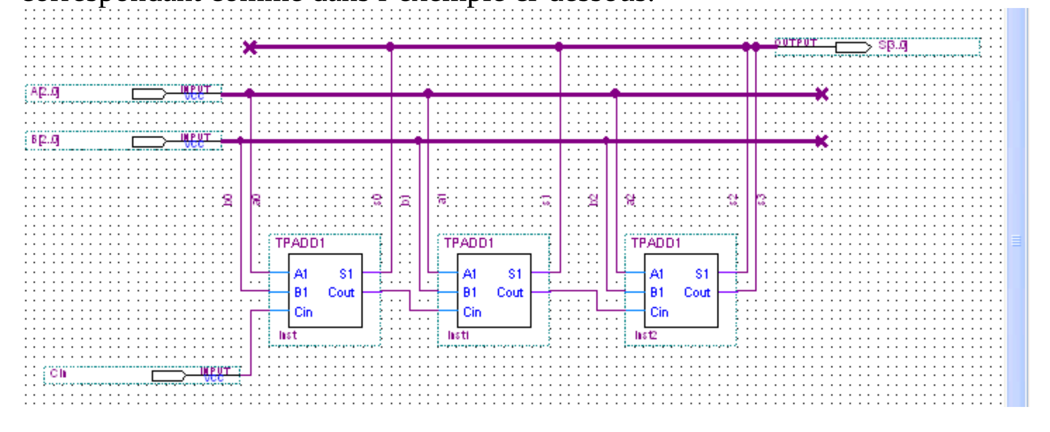
7.2 Description de l'instance ADD2X30

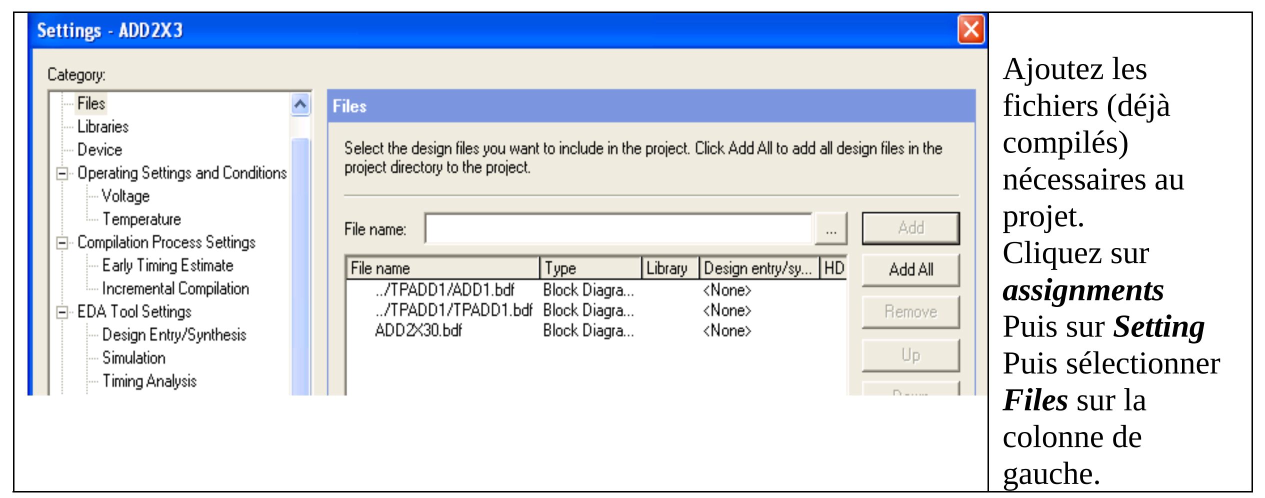
Nous allons maintenant décrire le fonctionnement de l'instance ADD2X30 grâce à une description schématique.
Cliquez sur l'instance avec le bouton droit : ***Create design file from selected block,*** choisir ***Schematic*** en s'assurant que l'option ***Add the new design file in the current project*** est activé.



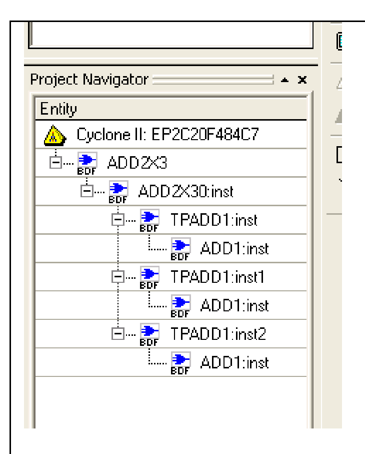
Le logiciel propose le nom du fichier généré sur la base du nom du block. Lors de l'ouverture de la nouvelle fenêtre graphique, les entrées et sorties sont automatiquement ajoutées.

Créez votre schéma à partir du symbole réalisé avec l'additionneur 1 bit. Il vous faut instancier 3 composants. La seule difficulté réside dans l'assignation du bon indice de vecteur de bit sur chaque instance. Pour isoler un indice d'un vecteur, il suffit de nommé le fil correspondant comme dans l'exemple ci-dessous.



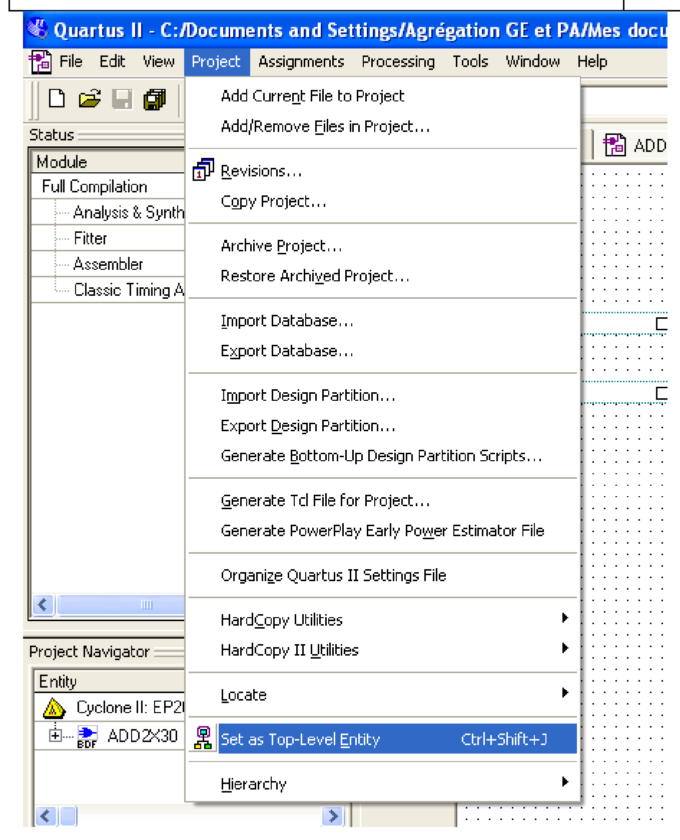


Ajoutez les fichiers (déjà compilés) nécessaires au projet.
Cliquez sur ***assignments***
Puis sur ***Setting***
Puis sélectionner ***Files*** sur la colonne de gauche.



Si vous le souhaitez vous pouvez simuler une entité sans simuler le design complet. Pour cela, il faut sélectionner l'entité à simuler dans le navigateur de projet.

Une fois l'entité ouverte il suffit de choisir Projet dans le menu et de sélectionner ***Set as Top-Level Entity***



Vérifiez le bon fonctionnement de cette entité avant d'aller plus loin.